

# From Graphs to Keyed Quantum Hash Functions

M. Ziatdinov

May 28, 2016

## Abstract

We present two new constructions of quantum hash functions: the first based on expander graphs and the second based on extractor functions and estimate the amount of randomness that is needed to construct them. We also propose a keyed quantum hash function based on extractor function that can be used in quantum message authentication codes and assess its security in a limited attacker model.

## 1 Introduction

Quantum hash functions are similar to classical (cryptographic) hash functions and their security is guaranteed by physical laws. However, their construction and applications are not fully understood.

Quantum hash functions were first implicitly introduced in Buhrman et al. [6] as quantum fingerprinting. Then Gavinsky and Ito [10] noticed that quantum fingerprinting can be used as cryptoprimitive. However, binary quantum hash function are not very suitable if we need group operations (and group is not  $\mathbb{Z}_{2^k}$ ). For example, several classical hash functions were proposed that use groups, e.g. by Charles et al. [7] and by Tillich and Zémor [18].

Ablayev and Ablayev [1] gave a definition and construction of non-binary quantum hash functions. Ziatdinov [21] showed how to generalize quantum hashing to arbitrary finite groups. Recently, Vasiliev [19] showed how quantum hash functions are connected with  $\epsilon$ -biased sets.

Quantum hash functions map a classical message into a Hilbert space. Such space should be as small as possible, so eavesdropper can't read a lot of information about classical message (this is guaranteed by physical laws as Holevo-Nayak's theorem states). But images of different messages should be as far apart as possible, so recipient can check that hash differ or not with high probability. We measure this distance using an absolute value of scalar product of hashes of different messages.

Informally speaking, to define a quantum hash function we need some random data. Then our input is mixed with this random data. Quantum parallelism allows us to do it in different subspaces simultaneously, so resulting hash is small. For example, random subsets suffice (for  $\mathbb{Z}_m$ ) [3], random codes suffice (for  $\mathbb{Z}_2^n$ ) [6], random automorphisms suffice (for any finite group) [21]. Vasiliev et al. [20] used some heuristics to find best subsets of  $\mathbb{Z}_m$ .

However, typically the amount of randomness that is needed to construct such quantum hash functions is large (about  $O(\log^2 |G|)$ ). We reduce amount of randomness needed to define quantum hash function to  $O(\log |G| \log \log |G|)$  in expander-based quantum hash function.

Extractor-based quantum hash function allows us to introduce a notion of keyed quantum hash function. It can be used, for example, in quantum message authentication codes. Unlike [5] and [4] we use classical keys and authenticate classical messages. Unlike [8] we authenticate whole messages, not single bits. However, our security analysis has only limited attacker.

It is known that walk on expander graph gives results very similar to random sampling. We show that walks on expander graphs give a quantum hash functions in section 4. Structure of these quantum hash functions is somewhat different from previous versions.

Extractor is a generalization of expander graph. In the section 6 we propose a keyed quantum hash function based on extractors and assess its security against limited attacker.

**Acknowledgements.** I thank Farid Abloyev, Alexander Vasiliev and Marco Carmosino for helpful discussions. A part of this research was done while attending a Special Semester Program on Computational and Proof Complexity (April-June 2016) organized by Chebyshev Laboratory of St.Petersburg State University in cooperation with Skolkovo Institute of Science and Technology and Steklov Institute of Mathematics at St.Petersburg. Partially supported by Russian Foundation for Basic Research, Grants 14-07-00557,

15-37-21160. The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

## 2 Definitions

Let us recall some basic definitions.

### 2.1 Statistics

We use a standard definition of the statistical distance.

**Definition 1** (Statistical distance, cited by Shaltiel [16]). *We say that two distributions  $F$  and  $G$  are  $\epsilon$ -close, if for every event  $A$ ,  $|\Pr[F \in A] - \Pr[G \in A]| \leq \epsilon$ .*

*The support of a distribution  $X$  is  $\text{Supp}(X) = \{x : \Pr[X = x] > 0\}$ .*

*The uniform distribution over  $\{0, 1\}^m$  is denoted by  $U_m$  and we say that  $X$  is  $\epsilon$ -close to uniform if it is  $\epsilon$ -close to  $U_m$ .*

*We denote that distribution  $F$  is  $\epsilon$ -close to distribution  $G$  by  $F \stackrel{\epsilon}{\approx} G$ .*

We also use a standard definition of the min-entropy.

**Definition 2** (Min-entropy, cited by Shaltiel [16]). *Let  $X$  be a distribution. The min-entropy of  $X$  is  $H_\infty(X) = \min_{x \in \text{Supp}(X)} \log \frac{1}{\Pr[X=x]}$ .*

### 2.2 Quantum model of computation

We use the following model of computation.

Recall that a qubit  $|\Psi\rangle$  is a superposition of basis states  $|0\rangle$  and  $|1\rangle$ , i.e.  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , where  $\alpha, \beta \in \mathbf{C}$  and  $|\alpha|^2 + |\beta|^2 = 1$ . So, qubit  $|\Psi\rangle \in \mathcal{H}^2$ , where  $\mathcal{H}^2$  is a two-dimensional Hilbert complex space.

Let  $s \geq 1$ . We denote  $2^s$ -dimensional Hilbert complex space by  $(\mathcal{H}^2)^{\otimes s}$ :

$$(\mathcal{H}^2)^{\otimes s} = \mathcal{H}^2 \otimes \mathcal{H}^2 \otimes \dots \otimes \mathcal{H}^2 = \mathcal{H}^{2^s}$$

We denote a state  $|a_1\rangle|a_2\rangle\dots|a_n\rangle$ , each  $a_i \in \{0, 1\}$ , by  $|i\rangle$ , where  $i$  is  $\overline{a_1 a_2 \dots a_n}$  in binary. For example, we denote  $|1\rangle|1\rangle|0\rangle$  by  $|6\rangle$ . Usually it is clear, which space this state belongs to.

Computation is done by multiplying a state by a unitary matrix:  $|\Psi_1\rangle = U|\Psi_0\rangle$ , where  $U$  is a unitary matrix:  $U^\dagger U = I$ ,  $U^\dagger$  is the conjugate matrix and  $I$  is the identity matrix.

The density matrix of a mixed state  $\{p_i, |\psi_i\rangle\}$  is a matrix  $\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|$ . A density matrix belongs to  $\text{Hom}((\mathcal{H}^2)^{\otimes s}, (\mathcal{H}^2)^{\otimes s})$ , the set of linear transformations from  $(\mathcal{H}^2)^{\otimes s}$  to  $(\mathcal{H}^2)^{\otimes s}$ .

At the end of computation state is measured by POVM (Positive Operator Valued Measure). A POVM on a  $(\mathcal{H}^2)^{\otimes s}$  is a collection  $\{E_i\}$  of positive semi-definite operators  $E_i : \text{Hom}((\mathcal{H}^2)^{\otimes m}, (\mathcal{H}^2)^{\otimes m}) \rightarrow \text{Hom}((\mathcal{H}^2)^{\otimes m}, (\mathcal{H}^2)^{\otimes m})$  that sums up to the identity transformation, i.e.  $E_i \succeq 0$  and  $\sum_i E_i = I$ . Applying a POVM  $\{E_i\}$  on a density matrix  $\rho$  results in answer  $i$  with probability  $\text{Tr}(E_i \rho)$ .

## 2.3 Character theory

**Definition 3** (Character of the group). *Let  $G$  be a group with unity  $e$  and operation  $\circ$ .*

*The character  $\chi : G \rightarrow \mathbb{C}$  of the group  $G$  is a homomorphism of  $G$  to  $\mathbb{C}$ : for any  $g, g' \in G$  it holds that  $\chi(g \circ g') = \chi(g)\chi(g')$ .*

## 2.4 Graphs

**Definition 4** (Expander graph, cited by Hoory et al. [14]). *Let the graph  $\Gamma = (V, E)$  with set of vertices  $V$  and set of edges  $E$  be fixed. Self-loops and multiple edges are allowed.*

*Graph  $\Gamma$  is the  $d$ -regular graph if all vertices have the same degree  $d$ ; i.e. each vertex is incident to exactly  $d$  edges.*

*Adjacency matrix of the graph  $A = A(\Gamma)$  is an  $n \times n$  matrix whose  $(u, v)$  entry is the number of edges between vertex  $u$  and vertex  $v$ .*

*Let  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  be eigenvalues of matrix  $A = A(\Gamma)$ , i.e. for some  $v_i$  it holds that  $Av_i = \lambda_i v_i$ . We refer to the eigenvalues of  $A(\Gamma)$  as the spectrum of the graph  $\Gamma$ .*

*Given a  $d$ -regular graph  $\Gamma$  with  $n$  vertices and spectrum  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  we denote  $\lambda(\Gamma) = \max\{|\lambda_2|, |\lambda_n|\}$ .*

*We call the graph  $\Gamma$  a  $(d, \lambda)$ -expander graph if  $\Gamma$  is  $d$ -regular and has  $\lambda(\Gamma) = \lambda$ .*

Every expander graph can be converted to a bipartite expander graph.

One can just take two copies of vertex sets and change original edges to go from one copy to another. Generalization of these bipartite expander graphs is extractor graphs. The extractor graph is a bipartite graph where size of components can be different. An extractor can also be defined in terms of function that maps pair of first component vertex and edge to second component vertex.

**Definition 5** ((Seeded) extractor, cited by Shaltiel [16]). *A function  $E : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  is a  $(k, \epsilon)$ -extractor if for every distribution  $X$  over  $\{0, 1\}^n$  with  $H_\infty(X) \geq k$ ,  $E(X, Y)$  is  $\epsilon$ -close to uniform (where  $Y$  is distributed like  $U_d$  and is independent of  $X$ ).*

Sometimes we use extractor functions that map one (arbitrary) set to other:  $E : G \times \{0, 1\}^d \rightarrow H$ . These functions can be thought of as bipartite graphs with vertices  $(G, H)$ . In this case we denote uniform distribution on  $H$  by  $U_H$ .

We also use extractors against quantum storage. Informally, their output is  $\epsilon$ -close to uniform and no quantum circuit operating on  $b$  qubits can distinguish output from uniform.

**Definition 6** (Extractor against quantum storage, cited by Ta-Shma [17]). *An  $(n, b)$  quantum encoding is a collection  $\{\rho(x)\}_{x \in \{0, 1\}^n}$  of density matrices  $\rho(x) \in (\mathcal{H}^2)^{\otimes b}$ .*

*A boolean test  $T$   $\epsilon$ -distinguishes a distribution  $D_1$  from a distribution  $D_2$  if  $|\Pr_{x_1 \in D_1}[T(x_1) = 1] - \Pr_{x_2 \in D_2}[T(x_2) = 1]| \geq \epsilon$ .*

*We say  $D_1$  is  $\epsilon$ -indistinguishable from  $D_2$  if no boolean POVM can  $\epsilon$ -distinguish  $D_1$  from  $D_2$ .*

*A function  $X : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  is a  $(k, b, \epsilon)$  strong extractor against quantum storage, if for any distribution  $X \subseteq \{0, 1\}^n$  with  $H_\infty(X) \geq k$  and every  $(n, b)$  quantum encoding  $\{\rho(x)\}$ ,  $U_t \circ E(X, U_t) \circ \rho(X)$  is  $\epsilon$ -indistinguishable from  $U_{t+m} \circ \rho(X)$ .*

### 3 Quantum hash functions

Informally, quantum hash function is a function that maps *large* classical input to a *small* quantum (hash) state such that two requirements are satisfied: (1) it is hard to restore input given the hash state and (2) it is easy to check with high probability that inputs for two quantum hash states are equal or different.

It is easy to meet the first requirement for a constant hash size. One can simply take a qubit  $|\Psi(w)\rangle = \alpha(w)|0\rangle + \beta|1\rangle$  and encode the input in a fractional part of  $\alpha$ . But then the second requirement is not satisfied.

It is easy to meet the second requirement for a hash size that is logarithmic in input size. One can simply map the input to the corresponding base state:  $|\Psi(i)\rangle = |i\rangle$ . However, then the first requirement is not satisfied.

Let us give the formal definition.

**Definition 7** (Quantum hash function, cited by Ablayev and Ablayev [2]). *For  $\delta \in (0, 1/2)$  we call a function  $\psi : X \rightarrow (\mathcal{H}^2)^{\otimes s}$  a  $\delta$ -resistant function if for any pair  $w, w'$  of different elements of  $X$  their images are almost orthogonal:*

$$|\langle \psi(w) | \psi(w') \rangle| \leq \delta. \quad (1)$$

*We call a map  $\psi : X \rightarrow (\mathcal{H}^2)^{\otimes s}$  an  $\delta$ -resistant  $(K; s)$  quantum hash function if  $\psi$  is a  $\delta$ -resistant function, and  $\log |X| = K$ .*

Quantum hash function maps inputs of length  $K$  to (quantum) outputs of length  $s$ . If  $K \gg s$  any attacker can't get a lot of information by Holevo-Nayak theorem [15].

The equality of two hashes can be checked using, for example, well-known SWAP-test [12].

All our hash functions have the following form:

$$|\psi(g)\rangle = \sum_{i=1}^t \chi(k_i(g)) |i\rangle, \quad (2)$$

where  $g$  is an element of some group  $G$ ,  $\{k_i, i = 1, \dots, t\}$ ,  $k_i : G \rightarrow H$  is a set of mappings from group  $G$  with operation  $\circ$  to group  $H$  with operation  $\bullet$  and  $\chi : H \rightarrow \mathbb{C}$  is a character of the group  $H$ .

For example, the group  $G$  can be thought of as  $Z_{2^n}$  with group operation  $+$ , then elements of  $G$  can be encoded as binary strings  $\{0, 1\}^n$  of length  $n$ .

### 3.1 Why groups?

We use groups in quantum hash functions of form (2), not just arbitrary sets, because groups have nice structure. We can combine elements of group and we can inverse them.

Several classical cryptoprimitives were proposed that use groups, e.g. by Charles et al. [7] and by Tillich and Zémor [18].

## 4 Expanders for Quantum Hashing

As noted in Section 1, randomly chosen parameters with high probability lead to a quantum hash function. We replace this process with random walk on expander graph that is known to be close to uniform sampling.

In this section we fix a group  $G$  with group operation  $\odot$  and unity  $e$ .

Let  $\Gamma = (V, E)$  be an extractor - i.e.  $d$ -regular graph with spectral gap  $\lambda$ . We label vertices  $V$  of graph  $\Gamma$  with messages (i.e. elements of group  $G$ ).

Let us randomly choose one vertex and perform a random walk of length  $t$  starting from it. Denote vertices that occurred in this walk by  $s_j$ . Parameter  $t$  depend on security parameter  $\epsilon$  of quantum hash function and we derive its value in theorem 1.

It is easy to note that such construction requires only  $td + \log |G|$  bits of randomness.

Let us define the expander quantum hash function.

**Definition 8.** *The expander quantum hash function  $\Psi_{\Gamma,t}(g)$  maps elements of  $G$  to unitary transformations in  $m$ -dimensional Hilbert space  $(\mathcal{H}^2)^{\otimes m}$ :*

$$\Psi_{\Gamma,t}(g) = \sum_{k=1}^t \chi(g \odot s_k) |k\rangle.$$

If we choose  $\Gamma$  and  $t$  appropriately,  $\Psi_{\Gamma,t}$  is a quantum hash function.

**Theorem 1.** *For any  $\delta \in (0; \frac{1}{2})$  the function  $\Psi_{\Gamma,t}$  is a  $\delta$ -resistant  $(\log |G|; \log t)$  quantum hash function if  $t > O(\frac{\log |G|}{\delta})$ .*

*Proof.* Let us fix some  $t$ .

$$\langle \Psi^\dagger(g) | \Psi_{\Gamma,t}(g') \rangle = \sum_{k=1}^t \langle \chi^*(g \odot s_k) | \chi(g' \odot s_k) \rangle = \left| \sum_{k=1}^t \chi(s_k^{-1} \odot g^{-1} \odot g' \odot s_k) \right|.$$

Denoting  $g'' = g^{-1} \odot g'$ , we get

$$\langle \Psi^\dagger(g) | \Psi_{\Gamma,t}(g') \rangle = \left| \sum_{k=1}^t \chi(s_k^{-1} \odot g'' \odot s_k) \right|,$$

and  $x_k = s_k^{-1} \odot g'' \odot s_k$  is also some random walk on graph  $\Gamma$ .

Let  $G$  be a weighted graph with eigenvalue gap  $\epsilon = 1 - \lambda$  and non-uniformity  $\nu$ . Let random walk on  $G$  starts in distribution  $q$  and has stationary distribution  $\pi$ . Then Chernoff bound for expander graphs [11] states that for any positive integer  $n$  and for any  $\gamma > 0$ :

$$\Pr \left[ \left| \sum_{i=1}^n f(x_i) - n \mathbf{E}_\pi f \right| \geq \gamma \right] \leq 4N_q \exp \left[ - \left( \frac{\gamma}{\|f\|_\infty} \right)^2 \frac{\epsilon}{20n} \right]. \quad (3)$$

Here we have graph weights  $w_{ij} = \frac{1}{d}$  for all  $i, j$  and  $w_x = 1$ , thus  $\nu = 1$  and  $\pi(x) = \frac{1}{V}$ . Initial distribution  $q$  is uniform distribution over  $G$ , therefore  $N_q = 1$ . Function  $f(x) = \chi(x)$  obviously has  $\|f\|_\infty \leq 1$ . We also bound (3) with some small probability, e.g.  $\frac{1}{|G|}$ . Then (3) becomes

$$\Pr \left[ \left| \sum_{i=1}^t f(x_i) - t \mathbf{E}_\pi f \right| \geq \gamma \right] \leq 4 \exp \left[ - \frac{\gamma^2 \epsilon}{20t} \right] \leq \frac{1}{|G|}.$$

Solving with respect to  $t$  gives us:

$$t \geq \frac{20}{(1 - \lambda)\delta} \ln(4|G|) = O(\log |G|).$$

If we make a random walk of length  $t = O(\log |G|)$ , we will get a quantum hash function with high probability.  $\square$

So, construction of this quantum hash function requires only  $O(\log |G|)$  bits of randomness if underlying expander graph is chosen carefully.

**Corollary 1.** *For all  $n$  and  $\delta \in (0; \frac{1}{2})$  there exist a  $\delta$ -resistant  $(\log n; \log t + 1)$  quantum hash function with  $t \geq \frac{160\sqrt{2}}{3\delta} \ln(4n)$ .*

*Proof.* We use Margulis construction [14] of  $(8; \frac{5\sqrt{2}}{8})$  expander graph with  $n^2$  vertices and character of group  $\mathbb{Z}_n^2$ .  $\square$

## 5 Extractors for Quantum Hashing

**Definition 9.** *Let  $\text{Ext} : G \times \{0, 1\}^d \rightarrow H$  be a  $(k; \epsilon)$  extractor function. Let  $t$  and  $s_i \in G, i \in \{1, \dots, t\}$  be parameters. We choose them in Theorem 2. Denote  $S = \{s_i\}$ .*



We define a quantum hash function  $\Psi$  based on extractor  $\text{Ext}$  as follows.

$$\Psi_{\text{Ext},t,S}(g) = \sum_{i=1}^t \sum_{j=1}^{2^d} \chi(\text{Ext}(g \circ s_i, j)) |j\rangle |i\rangle.$$

Intuitively, we start from several vertices and move along all incident edges simultaneously.

Parameters  $t, s_i$  depend on security parameter  $\epsilon$ . Let us choose it.

**Theorem 2.** *If  $\text{Ext}$  is a  $(k, \epsilon)$  extractor, parameter  $t > \frac{\log |H| + 1}{2\epsilon^2} \|\chi\|_\infty$  and  $s_i$  are chosen according to distribution  $X$  with  $H_\infty(X) \geq k$ , then  $\Psi_{\text{Ext}}$  is an  $\epsilon$ -resistant  $(n; d + \log t)$  quantum hash function.*

*Proof.* It is sufficient to prove that for any  $g' \neq g$

$$\begin{aligned} \left| \langle \Psi_{\text{Ext},t,S}(g) | \Psi_{\text{Ext},t,S}(g') \rangle \right| &= \left| \sum_{i=1}^t \sum_{j=1}^{2^d} \chi(\text{Ext}(g \circ s_i, j)^{-1} \bullet \text{Ext}(g' \circ s_i, j)) \right| \leq \\ &\leq \sum_{i=1}^t \sum_{j=1}^{2^d} |\chi(\text{Ext}(g \circ s_i, j)^{-1} \bullet \text{Ext}(g' \circ s_i, j))| < \epsilon. \end{aligned}$$

Define  $X_i$  to be a distribution of (random variable)  $s_i$ . Let  $Y_i$  be a random variable  $\mathbf{E}_{U_d}[\chi(\text{Ext}(X_i, U_d))]$ .

It is easy to see that  $Y_i \leq \|\chi\|_\infty = 1$ .

Then by Hoeffding's inequality:

$$\Pr \left[ \left| \frac{1}{t} \sum_{i=1}^t Y_i - \mathbf{E} \left[ \frac{1}{t} \sum_{i=1}^t Y_i \right] \right| \geq \epsilon \right] \leq 2 \exp \left( - 2t\epsilon^2 \right).$$

Bounding this probability by  $\frac{1}{|H|}$  and solving with respect to  $t$  gives

$$t \geq \frac{\log |H| + 1}{2\epsilon^2}.$$

□

Note that selecting parameters  $S$  requires  $O(\log |G| \times \log |H|)$  random bits.

**Corollary 2.** *For every  $\epsilon > 0$ ,  $\alpha > 0$  and all positive integers  $n, k$  there exist an  $\epsilon$ -resistant  $(n; \log t + d + 1)$  quantum hash function, where  $t \geq \frac{m+1}{2\epsilon^2}$ ,  $d = O(\log n + \log(1/\epsilon))$  and  $m \geq (1 - \alpha)k$ .*

*Proof.* Guruswami et al. [13] proved that for every  $\alpha > 0$  and all positive integers  $n, k$  and all  $\epsilon > 0$  there is an explicit construction of a  $(k; \epsilon)$  extractor  $E : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  with  $d = O(\log n + \log(1/\epsilon))$  and  $m \geq (1 - \alpha)k$ .

Quantum hash function  $\Psi_{E,t}$  is the required function.  $\square$

## 6 Keyed quantum hash functions

Classical message authentication codes (MAC) have wide range of applications. They are defined as a triple of algorithms:  $G$  that generates a key,  $S$  that uses the key and the message to generate a tag of the message, and  $V$  that uses the key, the message and the tag to verify message integrity.

Formally,  $G : 1^n \rightarrow K$ , where  $n$  is a security parameter and  $K$  is a set of all possible keys,  $S : K \times X \rightarrow T$ , where  $X$  is a set of messages and  $T$  is a set of tags and  $V : K \times X \times T \rightarrow \{\text{Acc}, \text{Rej}\}$ .

We require the following property for MAC to be a sound system:

$$\forall n, \forall x \in X : k = G(1^n), V(k, x, S(k, x)) = \text{Acc}, \quad (4)$$

i.e. that verifier always accepts a generated tag.

We also require that MAC is a secure system and for any adversary  $A$  that can query MAC:

$$\forall n, k \notin \text{Query}(A), (x, t) \leftarrow A(S), \Pr [V(k, x, t) = \text{Acc}] \leq \text{negl}(n), \quad (5)$$

i.e. any adversary that can query MAC outputs correct tag for some key that was not queried and some message with negligible probability.

One classical construction of MAC is hash-based MAC (also known as keyed hash functions). Basically, keyed hash function is a function  $H(k, x)$ , such that  $H(k, \cdot)$  is a cryptographic hash function for every  $k$ . It is easy to see that such function can be used as MAC.

With the same considerations as in Section 3, we define these algorithms to be the following.

**Definition 10.** *An  $(\epsilon, \delta)$  keyed quantum hash function is a quantum function  $S$ , such that*

A function  $S$  accepts a key  $k \in K$  and a message  $x \in X$  and outputs a quantum tag for  $x$ :  $S : K \times X \rightarrow T = (\mathcal{H}^2)^{\otimes t}$ .

We require soundness, i.e. tags should be different for different messages under the same key.

$$\forall k \in K, \forall x \in X, \forall y \neq x : \langle S(k, x) | S(k, y) \rangle < \epsilon.$$

For  $x = y$  we get  $\langle S(k, x) | S(k, x) \rangle = 1$ .

We also require unforgeability:

$$\forall k \in K, k \notin \text{Query}(A), (x, t) \leftarrow A(S), \Pr [\langle t | S(k, x) \rangle \geq \epsilon] \leq \delta,$$

where  $A$  is arbitrary attacker that can query  $S$  and  $(\text{Query})(A)$  is a set of queries made.

Informally, keyed quantum hash function outputs a tag for a message. If someone changes a message, then the verification step fails with high probability. If an attacker Eve can query a keyed quantum hash function, access to a function doesn't help her to forge a tag for some message with some (unqueried) key.

**Theorem 3.** Let us define an extractor-based keyed quantum hash function as follows. Let  $\text{Ext} : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  be a  $(k, b, \epsilon)$  extractor against  $b$  quantum storage and  $b > r(d + \log t)$ .

Then a function

$$\Psi_{\text{Ext}}(\text{key}, g) = \sum_{i=1}^t \sum_{j=1}^{2^d} \chi(\text{Ext}(g \circ \text{key} \circ s_i, j)) |j\rangle |i\rangle$$

is a  $(\epsilon; \epsilon + \epsilon^{2^s+1})$  keyed quantum hash function secure against an attacker  $A$  with access to  $r$  queries to  $\Psi_{\text{Ext}}$ .

*Proof.* We have to prove two claims. First, for any  $k, x$  and  $x' \neq x$ , it holds that  $\langle \Psi_{\text{Ext}}(k, x) | \Psi_{\text{Ext}}(k, x') \rangle < \epsilon$ . Second, for any attacker  $A$  and any  $k \notin \text{Query}(A)$  attacker output  $x, t$  such that  $\langle t | \Psi_{\text{Ext}}(k, x) \rangle \geq \epsilon$  with negligible probability.

The first claim is implied by Theorem 2.

To prove the second claim we note that access to hash function doesn't help attacker to output correct tag. Proof by contradiction. Suppose  $A$  to

be such attacker. Then we can distinguish between  $\text{Ext}(X, U_d)$  and  $U_m$  using a  $r(\log t + d)$  qubits. But  $r(\log t + d) < b$  that contradicts the fact that  $\text{Ext}$  is an extractor against  $b$  quantum storage.

Then attacker should output the tag without access to hash function. This is equal to outputting a state that is close to correct tag. Then the probability of correct guessing  $p$  is a ratio of the volume of sphere with radius  $\epsilon$  to the volume of the whole space:

$$p = \frac{c\epsilon^{2^s+1}}{c(1+\epsilon)^{2^s+1}} \leq \epsilon^{2^s+1}.$$

□

**Corollary 3.** *For all positive integers  $k, n$  and all  $c > 0$  there exist a  $(N^{-c}; 2N^{-c})$  keyed quantum hash function.*

*Proof.* De and Vidick [9] proved that for every  $\alpha, c > 0$  there exist an explicit  $(\alpha N, b, N^{-c})$  extractor  $E : \{0, 1\}^N \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  against  $b$  quantum storage with  $d = O(\log^4 n)$  and  $m = \Omega(\alpha N - b)$ . □

## 7 Open problems

Groups that we considered here and all constructions known to us use finite groups or sets and hash input strings of finite lengths.

**Problem 1.** *Can quantum hash functions be constructed for infinite groups?*

On the one hand, even one qubit can store arbitrary length binary string. On the other hand, the measurement of one qubit can't result in more than one classical bit of information.

And “dual” question about infinite strings.

**Problem 2.** *Can quantum hash functions work on infinite input strings (i.e.  $\{0, 1\}^*$ )?*

This problem seems to be easier, but it probably requires careful analysis.

Another interesting line of research would be improving keyed quantum hash function.

**Problem 3.** *Can keyed quantum hash function be secure against an attacker with unlimited number of queries?*

## References

- [1] F Ablayev and M Ablayev. On the concept of cryptographic quantum hashing. *Laser Physics Letters*, 12(12):125204, 2015. ISSN 1612-2011. doi: 10.1088/1612-2011/12/12/125204. URL <http://stacks.iop.org/1612-202X/12/i=12/a=125204?key=crossref.2f281c688485095be58bebb58f8dad75><http://iopscience.iop.org/article/10.1088/1612-2011/12/12/125204/meta>.
- [2] Farid Ablayev and Marat Ablayev. Quantum Hashing via Classical epsilon-universal Hashing Constructions. *arXiv*, pages 1–14, 2014. URL <http://arxiv.org/abs/1404.1503>.
- [3] Farid Ablayev and Alexander Vasiliev. On the Computation of Boolean Functions by Quantum Branching Programs via Fingerprinting. In *Electronic Colloquium on Computational Complexity*, volume 59, 2008.
- [4] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of quantum messages. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 449–458, 2002. ISSN 0272-5428. doi: 10.1109/SFCS.2002.1181969. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1181969>.
- [5] Howard Barnum. Quantum message authentication codes. *Quantum*, pages 1–18, 2001. URL <http://arxiv.org/abs/quant-ph/0103123>.
- [6] H Buhrman, R Cleve, J Watrous, and R de Wolf. Quantum fingerprinting. *Physical review letters*, 87(16):167902, 2001. ISSN 0031-9007. doi: 10.1103/PhysRevLett.87.167902.
- [7] Denis X. Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009. ISSN 09332790. doi: 10.1007/s00145-007-9002-x.
- [8] Marcos Curty and David J Santos. Quantum authentication of classical messages. 2013.
- [9] Anindya De and Thomas Vidick. Near-optimal extractors against quantum storage. 2009. URL <http://arxiv.org/abs/0911.4680>.

- [10] Dmitry Gavinsky and Tsuyoshi Ito. Quantum Fingerprints that Keep Secrets. page 28, 2010. ISSN 15337146. URL <http://arxiv.org/abs/1010.5342>.
- [11] D. Gillman. A Chernoff bound for random walks on expander graphs. *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 680–691, 1993. ISSN 0097-5397. doi: 10.1109/SFCS.1993.366819. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=366819>.
- [12] Daniel Gottesman and Isaac L Chuang. Quantum Digital Signatures. Technical report, 2001.
- [13] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. *Journal of the ACM*, 56(4):1–34, 2009. ISSN 10930159. doi: 10.1109/CCC.2007.38.
- [14] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006. ISSN 02730979. doi: 10.1090/S0273-0979-06-01126-8.
- [15] Ashwin V. Nayak. *Lower Bounds for Quantum Computation and Communication*. PhD thesis, California, Berkeley, 1999. URL <http://arxiv.org/abs/1011.1669><http://dx.doi.org/10.1088/1751-8113/44/8/085201>.
- [16] Ronen Shaltiel. An introduction to randomness extractors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6756 LNCS(PART 2):21–41, 2011. ISSN 03029743. doi: 10.1007/978-3-642-22012-8\_{\\_}2.
- [17] Amnon Ta-Shma. Short Seed Extractors Against Quantum Storage. *Proc. ACM STOC*, pages 401–408, 2009. ISSN 0097-5397. doi: 10.1145/1536414.1536470. URL <http://portal.acm.org/citation.cfm?doid=1536414.1536470>.
- [18] Jean-Pierre Tillich and Gilles Zémor. Group-theoretic hash functions. In *Algebraic Coding*, pages 90–110. Springer Berlin Heidelberg, 1994.

doi: 10.1007/3-540-57843-9{\\_}12. URL [http://link.springer.com/10.1007/3-540-57843-9{\\\_}12](http://link.springer.com/10.1007/3-540-57843-9{\_}12).

- [19] Alexander Vasiliev. Quantum Hashing for Finite Abelian Groups. pages 1–5, 2016. URL <http://arxiv.org/abs/1603.02209>.
- [20] Alexander Vasiliev, Marat Latypov, and Mansur Ziatdinov. Minimizing Collisions for Quantum Hashing. *International Journal of Applied Engineering Research*, pages 1–5, 2015.
- [21] Mansur Ziatdinov. Quantum hashing. Group approach. *Lobachevskii Journal of Mathematics*, (2), 2016. URL <http://arxiv.org/abs/1412.5135>.